

Open Research Online

The Open University's repository of research publications and other research outputs

Multimedia resource discovery

Book Section

How to cite:

Rüger, Stefan (2009). Multimedia resource discovery. In: Goker, Ayse and Davies, John eds. Information Retrieval: Searching in the 21st Century. Chichester: John Wiley and Sons, pp. 39–62.

For guidance on citations see [FAQs](#).

© 2009 Stefan Rueger

Version: Accepted Manuscript

Link(s) to article on publisher's website:

<http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470027622.html>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk

Multimedia resource discovery

Stefan Rüger

Knowledge Media Institute, The Open University

Walton Hall, Milton Keynes MK7 6AA, UK

s.rueger@open.ac.uk

June 14, 2009

1 Introduction

Resource discovery is more than just search: it is browsing, searching, selecting, assessing and evaluating, ie, ultimately accessing information. Giving users access to collections is one of the defining tasks of a library. For thousands of years the traditional methods of resource discovery have been facilitated by librarians: they create reference cards with meta-data that are put into catalogues (nowadays, databases); they also place the objects in physical locations that follow certain classification schemes and they answer questions at the reference desk.

The advent of digital documents has radically changed the organisation principles; now it is possible to *automatically* index and search document collections as big as the world-wide web *à la* Google and browse collections utilising author-inserted links. It is almost as if automated processing has turned the traditional library access paradigm upside down. Instead of searching meta-data catalogues in order to retrieve the document, web search engines search the full content of documents and retrieve their meta-data, ie, the location where documents can be found. Not all manual intervention has been abandoned, though. For example, the Yahoo! directory is an edited classification scheme of submitted web sites that are put into a browsable directory structure akin to library classification schemes.

Undoubtedly, however, it is the automated approaches that have made all the difference to the way the vast world-wide web can be used. While the automated indexing of text documents has been successfully applied to collections as large as the world-wide-web for over a decade now, multimedia indexing by content involves different, still less mature and less scalable technologies. Figure 1 shows a matrix of different search engines depending on the query (columns) and document repository (rows). Entry a) in this matrix corresponds to a traditional text search engine with a completely different technology than entry b), a system that allows you to express musical queries by humming a tune and that then plays the corresponding song; the three c) entries in Figure 1 correspond to a multi-modal video search engine allowing search by motion with example images and text queries, eg, *find me video shots of the Houses of Parliament in London zooming into its tower's clock face using the following example still image*; in contrast to this d) could be a search engine with a query text box that returns BBC Radio 4 discussions. Section 2 summarises different basic technologies involved in multimedia search.

Multimedia collections pose their very own challenges; for example, images and videos don't often come with dedicated reference cards or meta-data, and when they do, as in museum collections, their creation will have been expensive and time-consuming. Section 3 explores the difficulties and limitations of automatically indexing, labelling and annotating image and video content. It briefly discusses the inherent challenges of the semantic gap, polysemy, fusion and responsiveness.

	text	stills	sketch	speech	sound	humming	motion	query
								documents
a								text
c	c				e		c	video
								images
d								speech
						b		music
								sketches

entry e:

you roar like a lion
 and get a wildlife
 documentary

Figure 1: New search engine types

Even if all these challenges were solved, indexing sheer mass is no guarantee of a successful annotation either: While most of today’s inter-library loan systems allow access to virtually any book publication in the world — at least around 98m book entries in OCLC’s Worldcat database (OCLC 2008) and 3m entries from Bowker’s Books In Print — students and researchers alike seem to be reluctant to actually make use of this facility. On the other hand, the much smaller catalogue offered by Amazon appears to be very popular, presumably owing to added services such as subject categories; fault tolerant search tools; personalised services telling the customer what’s new in a subject area or what other people with a similar profile bought; pictures of book covers; media and customer reviews; access to the table of contents, to selections of the text and to the full-text index of popular books; and the perception of fast delivery. In the multimedia context Section 4 argues that automated added services such as visual queries, relevance feedback and summaries can prove useful for resource discovery in multimedia digital libraries. Sections 4.1 is about summarising techniques for videos, Section 4.2 exemplifies visualisation of search results, while Section 4.3 discusses content-based visual search modes such as query-by-example and relevance feedback.

Finally, Section 5 promotes browsing as resource discovery mode and looks at underlying techniques to automatically structure the document collection to support browsing.

2 Basic Multimedia Search Technologies

The current best practice to index multimedia collections is via the generation of a library card, ie, a dedicated database entry of meta-data such as author, title, publication year and keywords. Depending on the concrete implementation these can be found with SQL queries, text-search engines or XML query language, but all these search modes are based on text descriptions of some form and are agnostic to the structure of the actual objects they refer to, be it books, CDs, videos, newspaper articles, paintings, sculptures, web pages, consumer products etc.

The left column of the matrix of Figure 1 is underpinned by text search technology and requires the textual representation of the multimedia objects, an approach that I like to call *piggy-back text retrieval*. Other approaches are based on an automatic classification of multimedia objects and on assigning words from a fixed vocabulary. This can be a certain camera motion that can be detected in a video (zoom, pan, tilt, roll, dolly in and out, truck left and right, pedestal up and down, crane boom, swing boom etc); a genre for music pieces such as jazz, classics; a generic scene description in images such as inside/outside, people, vegetation, landscape, grass, city-view etc or specific object detection like faces and cars etc. These approaches are known as *feature classification* or *automated annotation*.

The type of search that is most commonly associated with multimedia is *content-based*: The basic idea is that still images, music extracts, video clips themselves can be used as

queries and that the retrieval system is expected to return ‘similar’ database entries. This technology differs most radically from the thousands-year-old library card paradigm in that there is no necessity for meta-data at all. In certain searches there is the desire to match not only the general type of scene or music that the query represents, but instead one and only one exact multimedia object. For example, you take a picture of a painting in a gallery and submit this as a query to the gallery’s catalogue in the hope of receiving the whole database record about this particular painting, and not a variant or otherwise similar exhibit. This is sometimes called *fingerprinting* or *known-item search*.

The rest of this section outlines these four basic multimedia search technologies.

2.1 Piggy-back text retrieval

Amongst all media types, TV video streams arguably have the biggest scope for automatically extracting text strings in a number of ways: directly from closed-captions, teletext or subtitles; automated speech recognition on the audio and optical character recognition for text embedded in the frames of a video. Full text search of these strings is the way in which most video retrieval systems operate, including Google’s latest TV search engine <http://video.google.com> or Blinkx-TV <http://www.blinkx.tv>. In contrast to television, for which legislation normally requires subtitles to assist the hearing impaired, videos stored on DVD don’t usually have textual subtitles. They have *subpicture* channels for different languages instead, which are overlayed on the video stream. This requires the extra step of optical character recognition, which can be done with a relatively low error rate owing to good quality fonts and clear background/foreground separation in the subpictures. In general, teletext has a much lower word error rate than automated speech recognition. In practice, it turns out that this does not matter too much as query words often occur repeatedly in the audio - the retrieval performance degrades gracefully with increased word error rates.

Web pages afford some context information that can be used for indexing multimedia objects. For example, words in the anchor text of a link to an image, a video clip or a music track, the file name of the object itself, meta-data stored within the files and other context information such as captions. A subset of these sources for text snippets are normally used in web image search engines.

Some symbolic music representations allow the conversion of music into text, such as MIDI files which contain a music representation in terms of pitch, onset times and duration of notes. By representing differences of successive pitches as characters one can, for example, map monophonic music to one-dimensional strings. A large range of different text matching techniques can be deployed, for example the edit distance of database strings with a string representation of a query. The edit distance between two strings computes the smallest number of deletions, insertions or character replacements that is necessary to transform one string into the other. In the case of query-by-humming, where a pitch tracker can convert the hummed query into a MIDI-sequence (Birmingham et al 2006), the edit distance is also able to deal gracefully with humming errors. Other techniques create fixed-length strings, so called *n*-grams, with windows that glide over the sequence of notes. The resulting strings can be indexed with a normal text search engine. This approach can also be extended to polyphonic music, where more than one note can be sounded at any one time (Doraisamy and Rüger 2003).

2.2 Automated annotation

Automatically annotating images with text strings is less straightforward. Methods attempting this task include dedicated machine vision models for particular words (such as “people” or “aeroplane”). However, the most popular and successful *generic* approaches are based

on classification techniques. This normally requires a large training set of images that have annotations and from these one can extract features, or characteristics (see Section 2.3), and somehow correlate these with the existing annotations of the training set. For example, images with tigers will have orange-black stripes and often green patches from surrounding vegetation, and their existence in an unseen image can in turn bring about the annotation “tiger”.

In particular, the following techniques have been studied: machine translation methods that link image regions (blobs) and words in the same way as corresponding words in two text documents written in different languages but otherwise of same contents (Duygulu et al 2002); co-occurrence models of low-level image features of tiled image regions and words (Mori et al 1999); cross-lingual information retrieval models (Jeon et al 2003, Lavrenko et al 2003); inference networks that connect image segments with words (Metzler and Manmatha 2004); probabilistic modelling with latent Dirichlet allocation (Blei and Jordan 2003), Bernoulli distributions (Feng et al 2004) or non-parametric density estimation with EMD kernels (Yavlin-sky et al 2005); support vector machine classification and relevance feedback (Izquierdo and Djordjevic 2005); information-theoretic semantic indexing (Magalhães and Rüger 2007); and simple scene-level statistics (Torralba and Oliva 2003). All these methods have in common that a controlled vocabulary of limited size (in the order of 500 more or less general terms) is used to annotate images based on a large training set.

Both those machine vision methods that model one particular item and the more generic machine learning methods that create individual models for every term of a limited vocabulary show a varying degree of success and, in general, relatively large error rates. As such these automated annotation methods are more suitable in the context of browsing or in conjunction with other search methods. If you want to “find shots of the front of the White House in the daytime with the fountain running” (TRECVID 2003, topic 124), then a query-by-example search in a large database may be solved quicker and better by emphasising those shots that were classified as “vegetation”, “outside”, “building” etc even though the individual classification may be wrong in a significant proportion of cases.

Machine learning methods are not limited to images at all. For example one can extract motion vectors from MPEG-encoded videos and use these to classify a video shot independently into categories such as “object motion from left to right”, “zoom in” etc. In contrast to the above classification tasks the extracted motion vector features are much more closely correlated to the ensuing motion label than image features are to text labels, and the corresponding learning task is much simpler.

Musical genre classification can be carried out on extracted audio-features that represent a performance by its statistics of pitch content, rhythmic structure and timbre texture (Tzanetakis and Cook 2002): timbre texture features are normally computed using short-time Fourier transform and Mel-frequency cepstral coefficients that also play a vital role in speech recognition; the rhythmic structure of music can be explored using discrete wavelet transforms that have a different time resolution for different frequencies; pitch detection, especially in polyphonic music, is more intricate and requires more elaborate algorithms. For details, see the work of Tolonen and Karjalainen (2000). Tzanetakis and Cook (2002) report correct classification rates of between 40% (rock) and 75% (jazz) in their experiments with 10 different genres.

2.3 Content-based retrieval

The query-by-example paradigm extracts features from the query, which can be anything from a still image, a video clip, humming etc, and compares these with corresponding features in the database. The idea is to return to the user those items in the multimedia database that are most similar to the query, ie whose features are most similar to the features of the query.

Figure 2 shows a typical architecture of such a system.

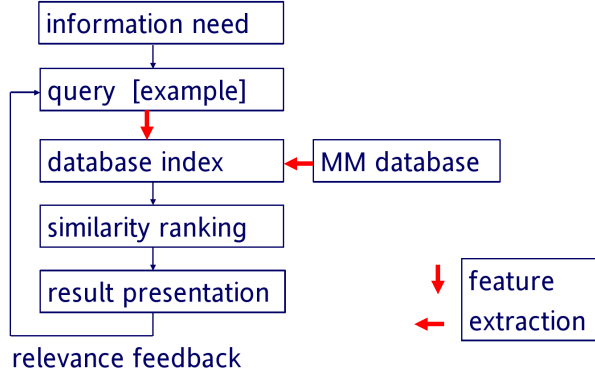


Figure 2: Content-based multimedia retrieval: generic architecture

The commonest way of indexing the visual content of images is by extracting low-level features, which represent colour usage, texture composition, shape and structure, localisation or motion. These representations are often real-valued vectors containing summary statistics, eg, in the form of histograms. Their respective distances act as indicators whether or not two images are similar with respect to this particular feature. Design and usage of these features can be critical, and there is a wide variety of them, as published by participants in the TRECVID conference (TRECVID 2003).

Once created, those features will allow the comparison and ranking of images in the database with respect to images submitted as a query, the *query-by-example* paradigm. Summary statistics are very crude indicators of similarity. For example an eight-bin histogram of intensity values of an image is a simple approximation of its brightness distribution, and many images do indeed share the same histogram. The image in Figure 3 shows a woman in the middle of bright column sculptures, but an image of a skier in snow is likely to have the same intensity histogram. The other disadvantage of global histograms computed over the whole image is that they lose any locality information. This can be alleviated by computing separate histograms for different areas of an image such as the centre, the border region, or in a number of tiles. Figure 3 is an example of a centre/border intensity histogram, where two histograms are computed for two different areas.

Normally, each multimedia document m gives rise to a number of low-level features $f_1(m), f_2(m), \dots, f_k(m)$, each of which would typically be a vector or numbers representing aspects like colour distribution, texture usage, shape encodings, musical timbre, pitch envelopes etc. Most systems accumulate the distances of these features to the corresponding features of the query q in order to define an overall distance:

$$D_w(m, q) = \sum_{i=1}^k w_i d_i(f_i(m), f_i(q)) \quad (1)$$

between multimedia documents m and a query q . Here $d_i(\cdot, \cdot)$ is a specific distance function between the vectors from the feature i , and $w_i \in \mathbb{R}$ is a weight for the importance of this feature. Note that the overall distance $D_w(m, q)$ is the number that is used to rank the multimedia documents in the database, and that the ones with the smallest distance to the query are shown to the user as query result, see Fig 2. Note also that the overall distance and hence the returned results crucially depend on the weight vector $w = (w_1, \dots, w_k)$. In most interfaces the user can either set the weights explicitly (as in the interface shown in Figure 8)

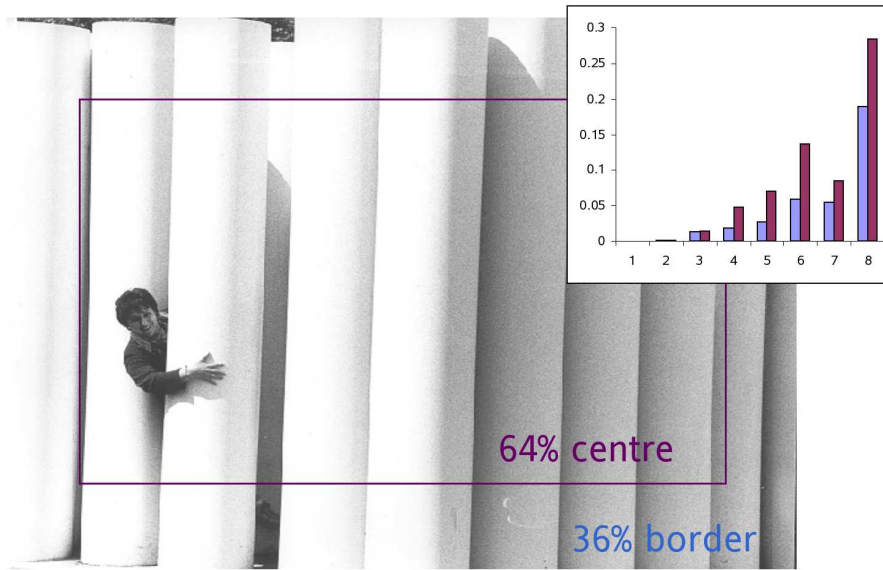


Figure 3: Centre-border intensity histogram of an image

or the system can change the weights implicitly if the user has given feedback on how well the returned documents fit their needs.

The architecture presented here is a typical albeit basic one; there are many variations and some radically different approaches that have been published in the past. A whole research field has gathered around the area of video and image retrieval as exemplified by the annual International ACM Conferences on Video and Image Retrieval (CIVR), Multimedia (ACM MM) and Multimedia Information Retrieval (MIR) and the TREC video evaluation workshop TRECVID; there is another research field around music retrieval, see the annual International Conference on Music Information Retrieval (ISMIR).

2.4 Fingerprinting

Multimedia fingerprints are unique indices in a multimedia database. They are computed from the contents of the multimedia objects, are small, allow the fast, reliable and *unique* location of the database record and are robust against degradation or deliberate change of the multimedia document that do not alter their human perception. Audio fingerprints of music tracks are expected to distinguish even between different performances of the same song by the same artist at perhaps different concerts or studios.

Interesting applications include services that allow broadcast monitoring companies to identify what was played, so that royalties are fairly distributed or programmes and advertisements verified. Other applications uncover copyright violation or, for example, provide a service that allows you to locate the meta-data such as title, artist and date of performance from snippets recorded on a (noisy) mobile phone.

Cano et al (2002) review some audio fingerprinting methods and Seo et al (2004) proposes an image fingerprinting technique.

3 Challenges of automated visual indexing

There are a number of open issues with the content-based retrieval approach in multimedia. On a perceptual level, those low-level features do not necessarily correlate with any high-level meaning the images might have. This problem is known as the *semantic gap*: imagine a scene

in which Bobby Moore, the captain of the English National Football team in 1966, receives the world cup trophy from Queen Elizabeth II; there is no obvious correlation between low-level colour, shape and texture descriptors and the high-level meaning of victory and triumph (or defeat and misery if you happened to support the West German team). Some of the computer vision methods go towards the bridging of the semantic gap, for example the ability to assign simple concrete labels to image parts such as “grass”, “sky”, “people”, “plates”. A consequent use of an ontology could explain the presence of higher-level concepts such as “barbecue” in terms of the simpler labels.

Even if the semantic gap could be bridged, there is still another challenge, namely *poly-semy*: images usually convey a multitude of meanings so that the query-by-example approach is bound to under-specify the real information need. Users who submit an image such as the one in Figure 3 could have a dozen different information needs in mind: “find other images with the same person”, “find images of the same art scene”, “find other bright art sculptures”, “find images with gradual shadow transitions”, ... It is these different interpretations that make further user feedback so important.

User feedback can change the weights in Equation (1), which represent the plasticity of the retrieval system. Hence, putting the user in the loop and designing a human-computer interaction that utilises the user’s feedback has been one of the main approaches to tackle these perceptual issues. Amongst other methods there are those that seek to reformulate the query (Ishikawa et al 1998) or those that weight the various features differently depending on the user’s feedback. Weight adaptation methods include cluster analysis of the images (Wood et al 1998); transposed files for feature selection (Squire et al 2000); Bayesian network learning (Cox et al 2000); statistical analysis of the feature distributions of relevant images and variance analysis (Rui et al 1998); and analytic global optimisation (Heesch and Rüger 2003). Some approaches give the presentation and placement of images on screen much consideration to indicate similarity of images amongst themselves (Santini and Jain 2000, Rodden et al 1999) or with respect to a visual query (Heesch and Rüger 2003).

On a practical level, the multitude of features assigned to images poses a *fusion problem*; how to combine possibly conflicting evidence of two images’ similarity? There are many approaches to carry out fusion, some based on labelled training data and some based on user feedback for the current query (Aslam and Montague 2001, Bartell et al 1994, Shaw and Fox 1994, Yavlinsky et al 2004).

There is a *responsiveness problem*, too, in that the naïve comparison of query feature vectors to the database feature vectors requires a linear scan through the database. Although the scan is eminently scalable, the practicalities of doing this operation can mean an undesirable response time in the order of seconds rather than the 100 milli-seconds that can be achieved by text search engines. The problem is that high-dimensional tree structures tend to collapse to linear scans above a certain dimensionality (Weber et al 1998). As a consequence, some approaches for fast nearest-neighbour search use compression techniques to speed up the disk access of linear scan as in (Weber et al 1998) using VA-files; or they approximate the search (Nene and Nayar 1997, Beis and Lowe 1997); decompose the features componentwise (de Vries et al 2002, Aggarwal and Yu 2000) saving access to unnecessary components; or deploy a combination of these (Müller and Henrich 2004, Howarth and Rüger 2005).

4 Added Services

4.1 Video summaries

Even if the challenges of the previous section were all solved and if the automated methods of Section 2 enabled a retrieval process with high precision (proportion of the retrieved items

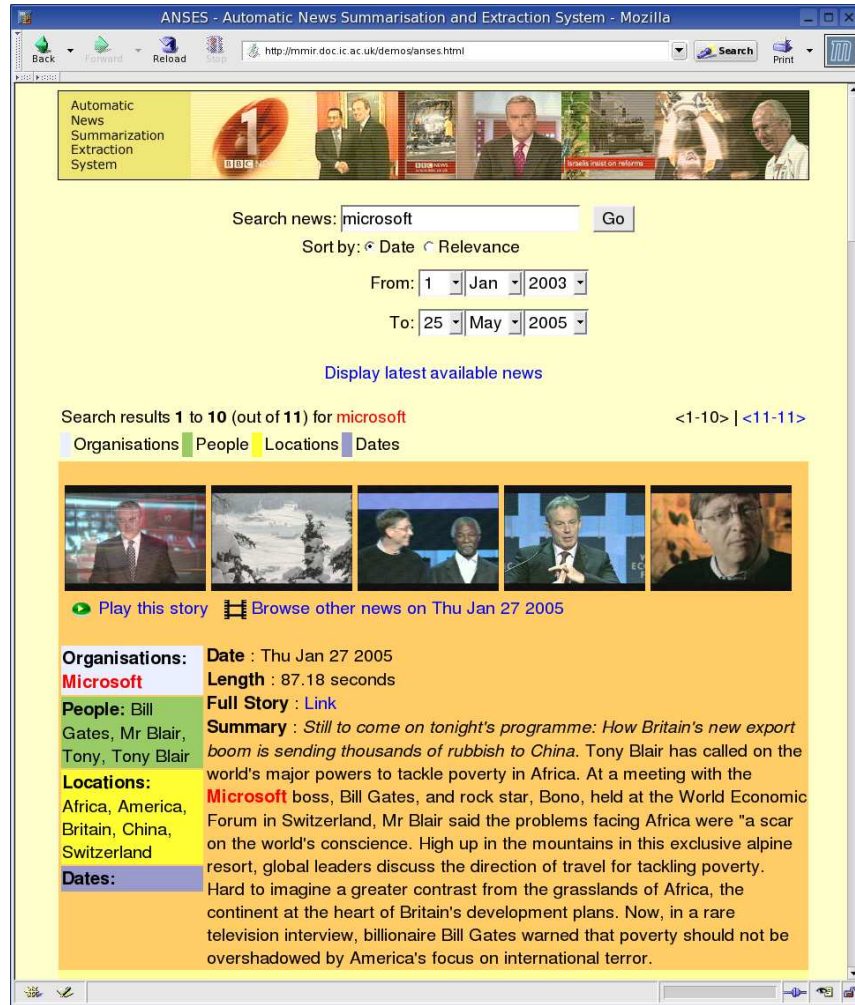


Figure 4: News search engine interface

that are relevant) and high recall (proportion of the relevant items that are retrieved) it would still be vital to present the retrieval results in a way so that the users can quickly decide to which degree those items are relevant to them.

Images are most naturally displayed as thumbnails, and their relevance can quickly be judged by users. Presenting and summarising videos is a bit more involved. The main metaphor used for this is that of a *storyboard* that contains *keyframes* with some text about the video. Several systems exist that summarise news stories in this way, most notably Informedia (Christel et al 1999) and Físchlár (Smeaton et al 2004). The Informedia system devotes much effort to added services such as face recognition and speaker voice identification allowing retrieval of the appearance of known people. Informedia also provides alternative modes of presentation, eg, through film skims or by assembling ‘collages’ of images, text and other information (eg, maps) sourced via references from the text (Christel and Warmack 2001). Físchlár’s added value lies in the ability to personalise the content (with the user expressing like or dislike of stories) and in assembling lists of related stories and recommendations.

Our very own TV news search engine ANSES (Pickering et al 2003, Pickering 2004) records the main BBC evening news along with the sub-titles, indexes them, breaks the video stream into shots (defined as those video sequences that are generated during a continuous operation of the camera), extracts one key-frame per shot, automatically glues shots together to form news stories based on an overlap in vocabulary in the sub-titles of adjacent shots (using lexical

chains), and assembles a story-board for each story. Stories can be browsed or retrieved via text searches. Fig 4 shows the interface of ANSES. We use the natural language toolset GATE (Cunningham 2002) for automated discovery of organisations, people, places and dates; displaying these prominently as part of a storyboard as in Figure 4 provides an instant indication of what the news story is about. ANSES also displays a short automated textual extraction summary, again using lexical chains to identify the most salient sentences. These summaries are never as informative as hand-made ones, but users of the system have found them crucial for judging whether or not they are interested in a particular returned search result.

Dissecting the video stream into shots and associating one keyframe along with text from subtitles to each shot has another advantage: A video collection can essentially be treated as an image collection, where each, possibly annotated image acts as entry point into the video.

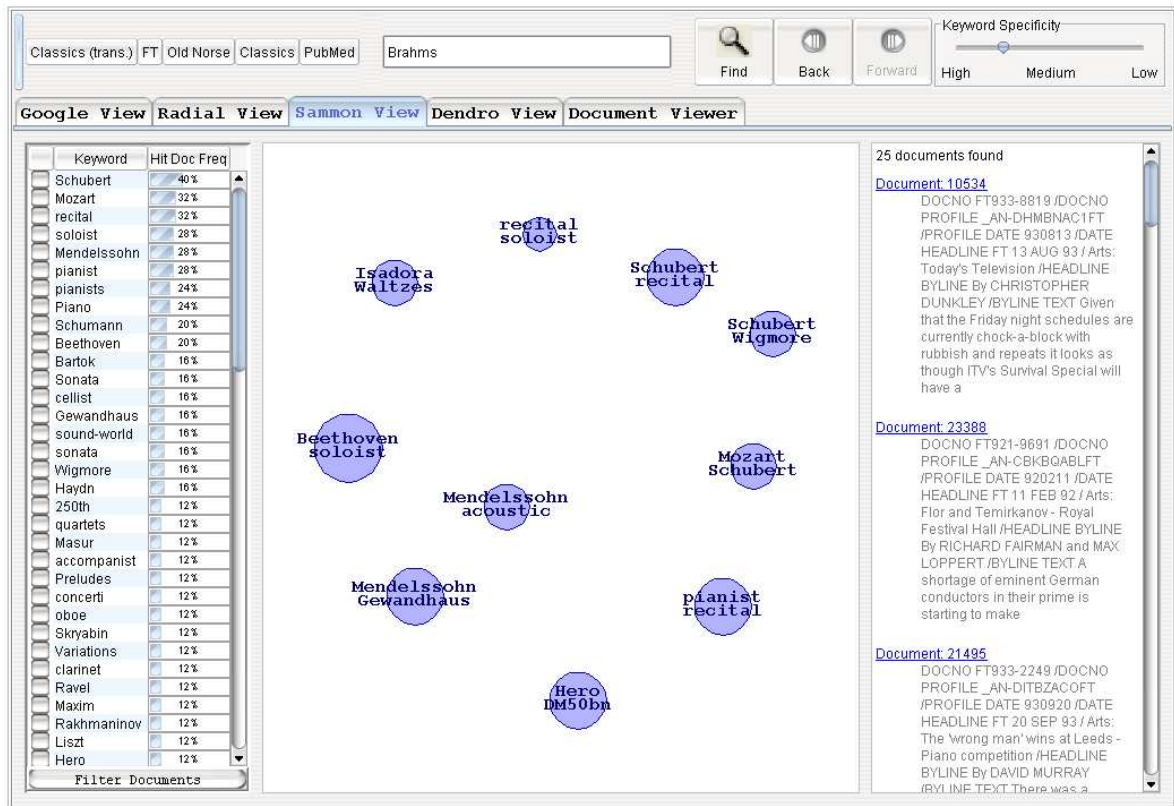


Figure 5: Sammon map for cluster-guided search

4.2 New Paradigms in Information Visualisation

The last decade has witnessed an explosion in interest in the field of information visualisation, (Hemmje et al 1994, Ankerst et al 1996, Card 1996, Shneiderman et al 2000, Börner 2000). Here we present three new visualisation paradigms, based on our earlier design studies (Au et al 2000, Carey et al 2003). These techniques all revolve around a representation of documents in the form of bag-of-words vectors, which can be clustered to form groups. We use a variant of the buckshot clustering algorithm for this. Basically, the top, say, 100 documents that were returned from a query are clustered via hierarchical clustering to initialise document centroids for k -means clustering that puts all documents returned by a query into groups. Another common element of our visualisations is the notion of *keywords* that are specific

to the returned set of documents. The keywords are computed using a simple statistic; for details see (Carey et al 2003). The new methods are:

Sammon Cluster View. This paradigm uses a Sammon map to generate a two dimensional screen location from a many-dimensional vector representing a cluster centroid. This map is computed using an iterative gradient search (Sammon 1969) while attempting to preserve the pairwise distances between the cluster centres. Clusters are thus arranged so that their mutual distances are indicative of their relationship. The idea is to create a visual landscape for navigation. Fig 5 shows an example of such an interface. The display has three panels, a scrolling table panel to the left, a graphic panel in the middle and a scrolling text panel to the right that contains the traditional list of returned documents as hotlinks and snippets. In the graphic panel each cluster is represented by a circle and is labelled with its two most frequent keywords. The radius of the circle represents the cluster size. The distance between any two circles in the graphic panel is an indication of the similarity of their respective clusters - the nearer the clusters, the more likely the documents contained within will be similar. When the mouse passes over the cluster circle a tool-tip box in the form of a pop-up menu appears that allows the user to select clusters and *drill down*, ie, re-cluster and re-display only the documents in the selected clusters. The back button undoes this process and climbs up the hierarchy (*drill up*). The table of keywords includes box fields that can be selected. At the bottom of the table is a filter button that makes the scrolling text window display only the hot-links and snippets from documents that contain the selected keywords.

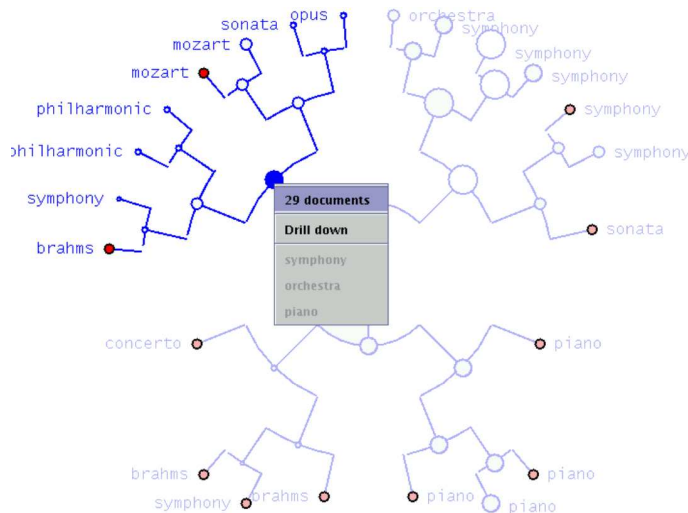


Figure 6: Dendro Map - A plane-spanning binary tree (query “Beethoven”)

Dendro Map Visualisation. The Dendro Map visualisation represents documents as leaf nodes of a binary tree that is output by the buckshot clustering algorithm. With its plane-spanning property and progressive shortening of branches towards the periphery, the Dendro Map mimics the result of a non-Euclidean transformation of the plane as used in hyperbolic maps without suffering from their computational load. Owing to spatial constraints, the visualisation depth is confined to five levels of the hierarchy with nodes of the lowest level representing either documents or subclusters. Different colours facilitate visual discrimination between individual documents and clusters. Each lowest level node is labelled with the most frequent keyword of the subcluster or document. This forms a key component of the Dendro Map as it gives the user the cues needed for navigating through the tree. As the user moves the mouse pointer over an internal node, the internal nodes and branches of the associated subcluster change colour from light blue to dark blue while the leaf nodes, ie, document representations, turn bright red. As in the Sammon Map, a tool-tip window provides additional

information about the cluster and can be used to display a table with a list of keywords associated with the cluster. The user may drill down on any internal node. The selected node will as a result replace the current root node at the center and the entire display is re-organized around the new root. The multi-level approach of the Dendro Map allows the user to gain a quick overview over the document collection and to identify promising subsets.

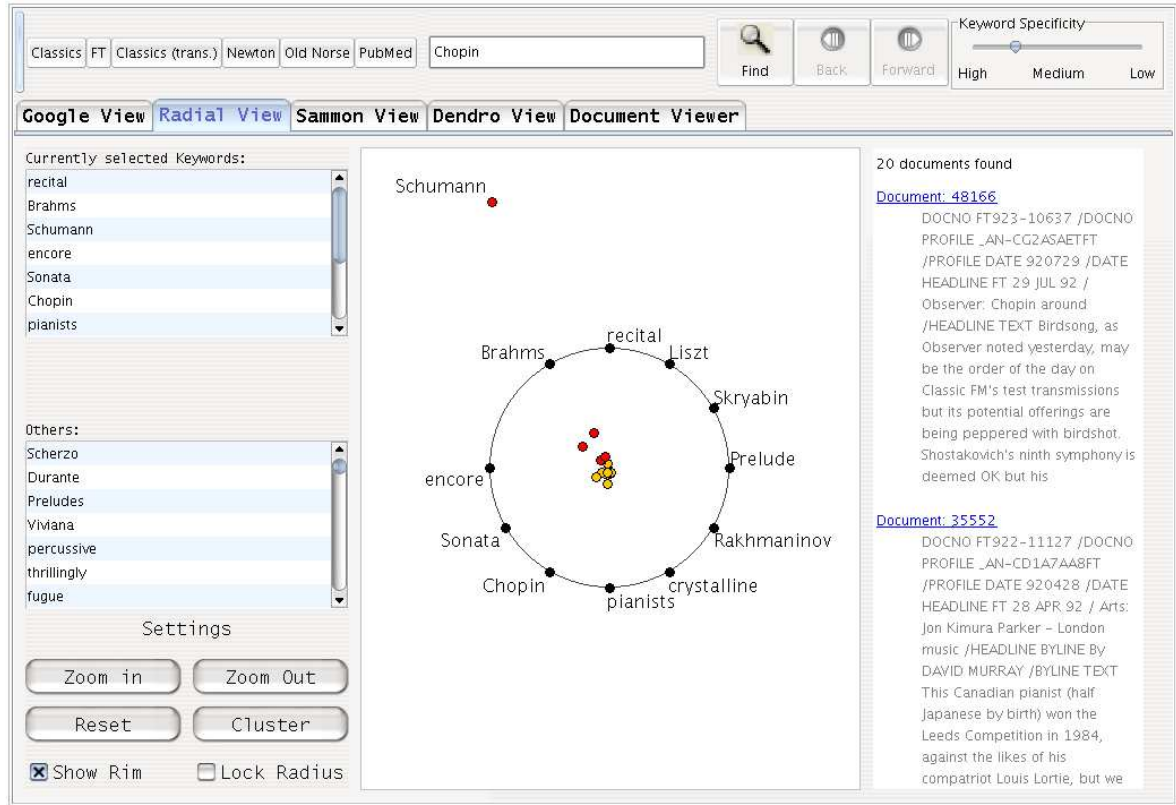
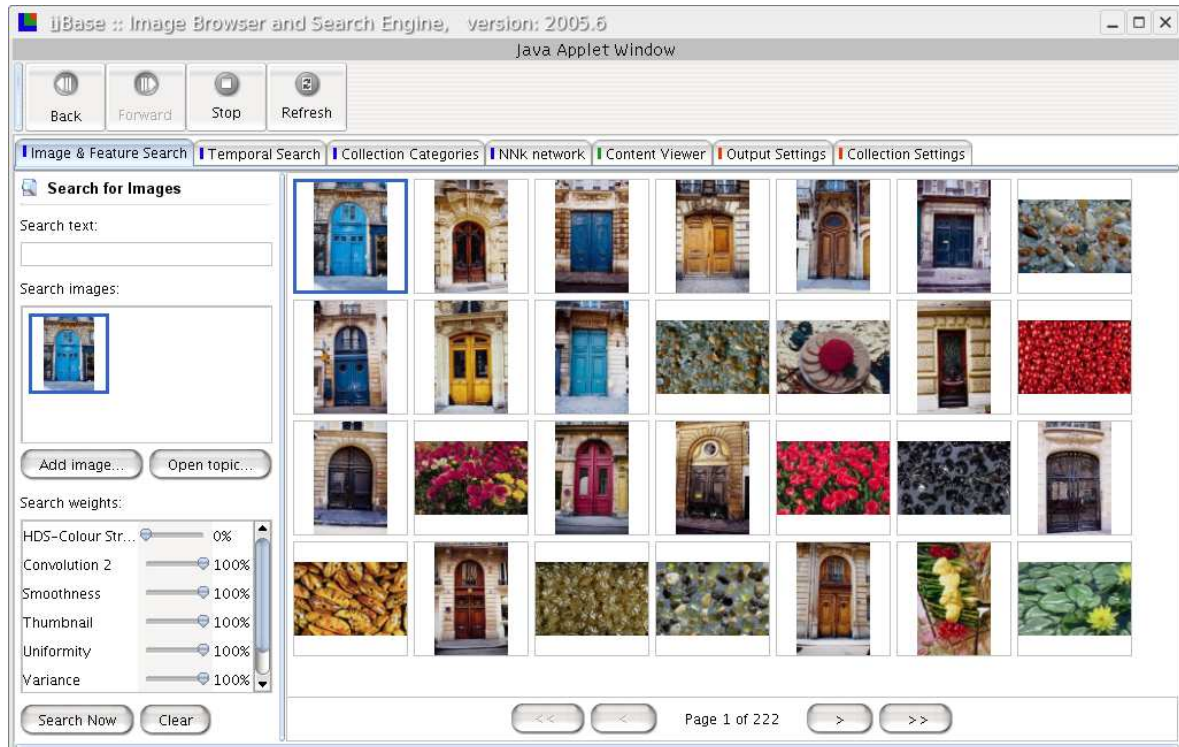
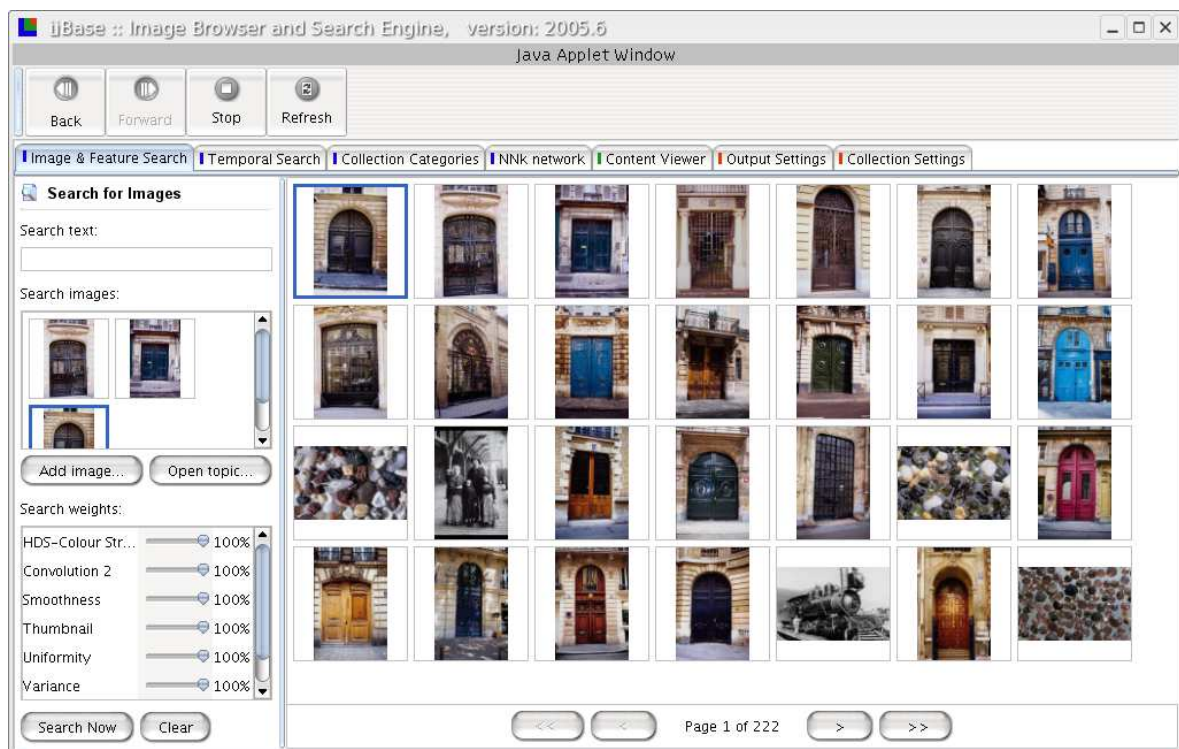


Figure 7: Radial Visualisation

Radial Interactive Visualisation. Radial (Figure 7) is similar to VIBE (Korfhage 1991), to Radviz (Hoffman et al 1999) and to Lyberworld (Hemmje et al 1994). It places the keyword nodes round a circle, and the position of the document dots in the middle depend on the force of invisible springs connecting them to keyword nodes: the more relevant a keyword for a particular document, the stronger its spring pulls on the document. Hence, we make direct use of the bag-of-words representation without explicit clustering. Initially, the twelve highest ranking keywords are displayed in a circle. The interface lets the user move the keywords, and the corresponding documents follow this movement. This allows the user to manually cluster the documents based on the keywords they are interested in. As the mouse passes over the documents, a bubble displays a descriptive piece of text. The location of document dots is not unique owing to dimensionality reduction, and there may be many reasons for a document to have a particular position. To mitigate this ambiguity in Radial the user can click on a document dot, and the keywords that affect the location of document are highlighted. A choice of keywords used in the display can be exercised by clicking on two visible lists of words. Zoom buttons allow the degree of projection to be increased or reduced so as to distinguish between documents around the edges of the display or at the centre. The Radial visualisation appears to be a good interactive tool to structure the document set according to one's own preferences by shifting keywords around in the display.



(a) Query by example (left panel) with initial results in the right panel



(b) A new query made of three images from (a) results in many more dark-door images

Figure 8: Visual search for images of dark doors starting with a bright-door example

Unified Approach. The integration of the paradigms into one application offers the possibility of browsing the same result set in several different ways simultaneously. The cluster-based visualisations give a broader overall picture of the result, while the Radial visualisation allows the user to focus on subsets of keywords. Also, as the clusters are approximations that highlight particular keywords, it may be useful to return to the Radial visualisation and examine the effect of these keywords upon the whole document set. The Radial visualisation will perhaps be more fruitful if the initial keywords match the user’s area of interest. The Sammon Map will let the user dissect search sets and re-cluster subsets, gradually homing in on target sets. This interface was developed within the joint NSF-EC project CHLT (<http://www.chlt.org>); it was evaluated from a human-computer-interaction point of view with encouraging results (Chawda et al 2005) and has proven useful in real-world multilingual scholarly collections (Rydberg-Cox et al 2004).

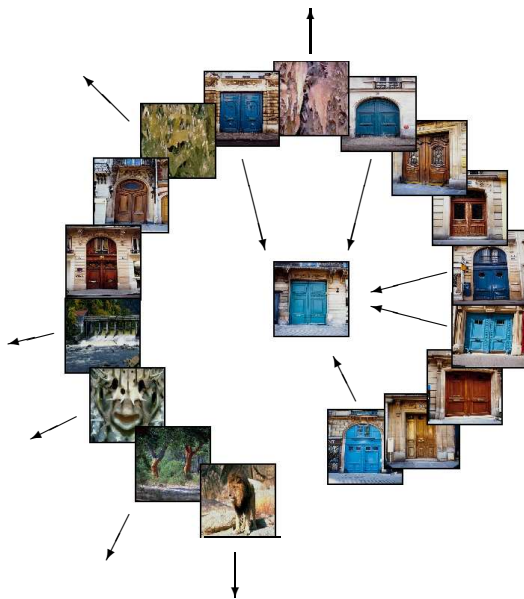


Figure 9: A relevance feedback model

4.3 Visual search and relevance feedback

The visual query-by-example paradigm discussed in Section 3 gives rise to relatively straightforward interfaces; an image is dragged into a query box, or, eg, specified via a URL, and the best matching images are displayed in a ranked list to be inspected by the user, see Fig 8(a). A natural extension of such an interface is to offer the selection of relevant results as new query elements. This type of relevance feedback, a.k.a. *query point moving*, is shown in Fig 8(b).

One other main type of relevance feedback, *weight space movement*, assumes that the relative weight of the multitude of features that one can assign to images (eg, structured meta-data fields such as author, creation date and location; low-level visual features such as colour, shape, structure and texture; free-form text) can be learned from user feedback. Of the methods mentioned in Section 3 our group chose analytic weight updating as this has a very small execution time. The idea is that users can specify the degree to which a returned image is relevant to their information needs. This is done by having a visual representation; the returned images are listed in a spiral, and the distance of an image to the centre of the screen is a measure of the relevance that the search engine assigns to a specific image. Users can now move the images around with the mouse or place them in the centre with a left

mouse click and far away with a right click. Fig 9 shows this relevance feedback model. We evaluated the effectiveness of negative feedback, positive feedback and query point moving, and found that combining the latter two yields the biggest improvement in terms of mean average precision (Heesch and Rüger 2003).

A new and relatively unexplored area of relevance feedback is the exploitation of social context information. By looking not only at the behaviour and attributes of the user, but also his past interactions and also the interactions of people he has some form of social connection with could yield useful information when determining whether search results are relevant or not. Browsing systems could recommend data items based on the actions of a social network instead of just a single user, using more data to yield better results.

The use of such social information is also becoming important for multimedia meta data generation, particular in the area of folksonomies where the feedback of users actively produces the terms and taxonomies used to describe the media in the system instead of using a pre-determined, prescribed dictionary (Voss 2007). This can be seen being effectively used in online multimedia systems such as Flickr (<http://www.flickr.com>) and del.icio.us (<http://del.icio.us>).

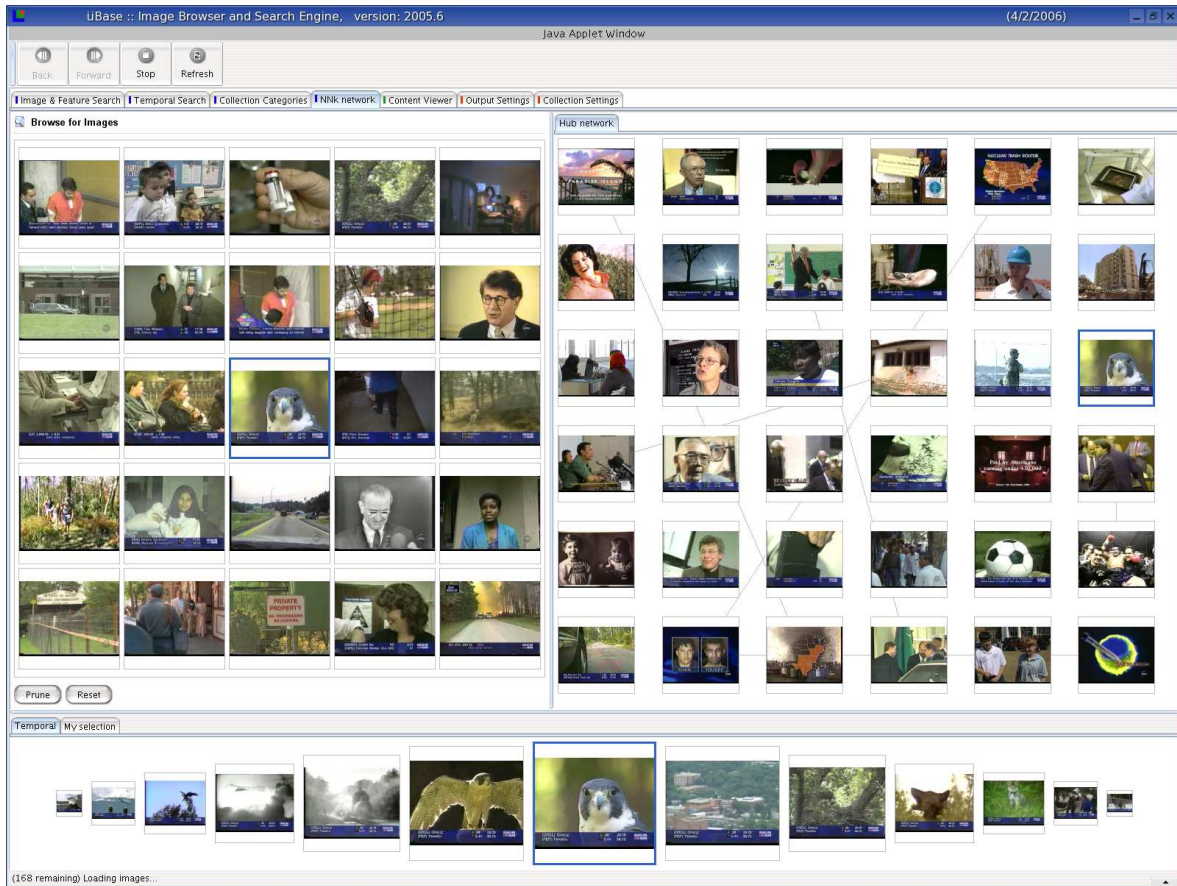
5 Browsing: lateral and geo-temporal

The idea of representing text documents in a nearest-neighbour network was first presented by Croft and Parenty (1985), albeit, as an internal representation of the relationships between documents and terms, not for browsing. Document networks for interactive browsing were identified by Cox (1992 and 1995). Attempts to introduce the idea of browsing into content-based image retrieval include Campbell’s work (2000); his ostensive model retains the basic mode of query based retrieval but in addition allows browsing through a dynamically created local tree structure. Santini and Jain’s *El niño* system (2000) is another attempt to combine query-based search with browsing. The system tries to display configurations of images in feature space such that the mutual distances between images are preserved as well as possible. Feedback is given in the same spirit as in Fig 9 by manually forming clusters of images that appear similar to the user. This in turn results in an altered configuration with potentially new images being displayed.

Other network structures that have increasingly been used for information visualisation and browsing are Pathfinder networks (Dearholt and Schvaneveldt 1990). They are constructed by removing redundant edges from a potentially much more complex network. Fowler et al (1992) use Pathfinder networks to structure the relationships between terms from document abstracts, between document terms and between entire documents. The user interface supports access to the browsing structure through prominently marked high-connectivity nodes.

Our group (Heesch and Rüger 2004) determines the nearest neighbour for the image under consideration (which we call the *focal* image) for *every* combination of features. This results in a set of what we call *lateral neighbours*. By calculating the lateral neighbours of all database images, we generate a network that lends itself to browsing. Lateral neighbours share some properties of the focal image, but not necessarily all. For example, a lateral neighbour may share text annotations with the focal image, but no visual similarity with it at all, or it may have a very similar colour distribution, but no structural similarity, or it may be similar in all features except shape, etc. As a consequence, lateral neighbours are deemed to expose the polysemy of the focal image. Hence, when they are presented, the user may then follow one of them by making it the focal image and explore its lateral neighbours in turn. The user interaction is immediate, since the underlying network was computed offline.

We provide the user with entry points into the database by computing a representative set of images from the collection. We cluster high-connectivity nodes and their neighbours

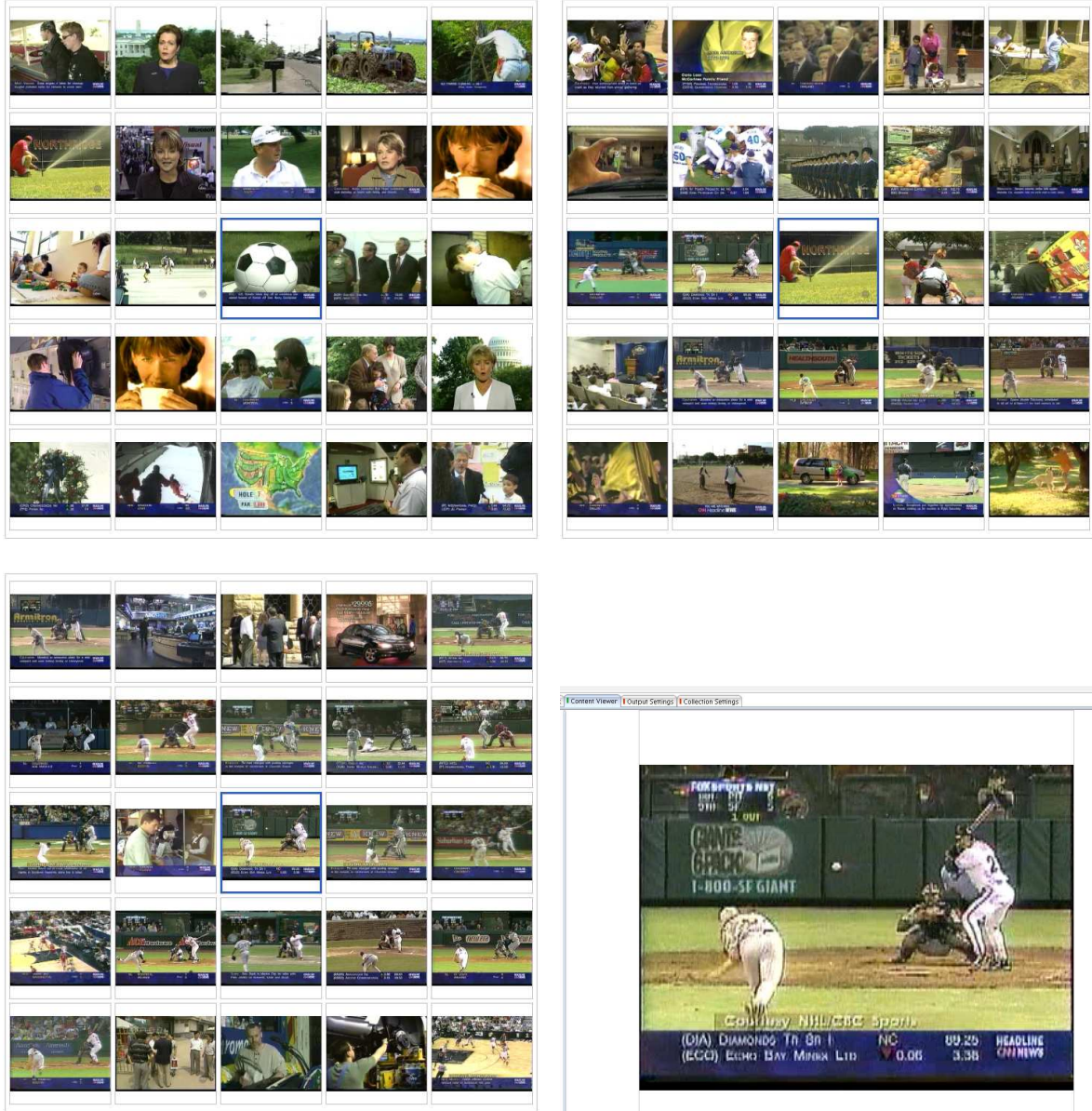


(a) Initial visual summary of the database (right panel) from which the user chooses the falcon, its nearest lateral neighbours are then displayed in the left panel.



(b) Clicking on any image will make it the centre of the nearest neighbours panel and display its associated lateral neighbours around it.

Figure 10: Lateral browsing for an image “from behind the pitcher in a baseball game...”



Starting with the football image (upper left) from the database overview, one of its lateral neighbours is an image of a lawn with a sprinkler; when this is made the focal image (upper right) there are already images from baseball scenes. Clicking on one of them (lower left) reveals that there are more of this kind; they can be enlarged and the corresponding video played in the “viewer tab” (lower right).

Figure 11: Alternative ways to browse for images “from behind the pitcher ...”

up to a certain depth using the Markov chain clustering algorithm (van Dongen 2000), which has robust convergence properties and allows one to specify the granularity of the clustering. The clustering result can be seen as a image database summary that shows highly-connected nodes with far-reaching connections. The right panel of Fig 10(a) is such a summary for our TRECVID (2003) database. The user may select any of these images as an entry point into the network. Clicking on an image moves it into the centre around which the lateral neighbours are displayed, see the nearest-neighbour panel on the left side of fig 10(a). If the size of the lateral-neighbour set is above a certain threshold the actual number of images displayed is reduced to the most salient ones.

If a user wanted to find “video shots from behind the pitcher in a baseball game as he throws a ball that the batter swings at” (TRECVID 2003, topic 102) then they might explore the database in Fig 10 by clicking on the falcon image. The hope is that the colour of a baseball field is not far off from the green colour of that image. The resulting lateral neighbours, displayed in the left panel of Fig 10(a), do not contain the desired scene. However, there is an image of a sports field. Making that the focal image, as seen in the left part of fig 10(b), reveals it has the desired scene as a lateral neighbour. Clicking that will unearth a lot more images from baseball fields, see the right side of Fig 10(b). The network structure, a bit of lateral thinking and three mouse clicks have brought the desired result.

In the same way, and again with only three clicks, one could have started from the football image in the database overview to find “video shots from behind the pitcher in a baseball game as he throws a ball that the batter swings at”. Heesch (2005) has shown that this is no coincidence; lateral-neighbour networks computed in this way have the so-called *small world property* (Watts and Strogatz 1998) with only 3–4 degrees of separation even for the large TRECVID (2003) database that contains keyframes from 32,000 video shots. Lateral browsing has proven eminently successful for similar queries (Heesch et al 2003).

Geo-temporal browsing takes the idea of timelines and automatically generated maps, eg as offered in the Perseus Digital Library (Crane 2005), a step further. It integrates the idea of browsing in time and space with a selection of events through a text search box. In this way, a large newspaper or TV news collection can be made available through browsing based on what happened where and when as opposed to by keyword only.

The interface in Fig 12 is a design study in our group that allows navigation within a large news event dataset along three dimensions: time, location and text subsets. The search term presents a text filter. The temporal distribution can be seen in lower part. The overview window establishes a frame of reference for the user’s region of interest. In principle, this interface could implement new zooming techniques, eg speed-dependent automatic zooming (Cockburn and Savage 2003), and link to a server holding a large quantity of maps such as National Geographic’s MapMachine (<http://plasma.nationalgeographic.com/mapmachine/> as of May 2005) with street-level maps and aerial photos.

6 Summary

This chapter has introduced basic concepts of multimedia resource discovery technologies for a number of different query and document types; these were the piggy-back text search, automated annotation, content-based retrieval and fingerprinting. The paradigms we have discussed include summarising complex multimedia objects such as TV news, information visualisation techniques for document clusters, visual search by example, relevance feedback and methods to create browsable structures within the collection. These exploration modes share three common features: they are automatically generated, depend on visual senses and interact with the user of the multimedia collections.

Multimedia resource discovery has its very own challenges in the semantic gap, in poly-

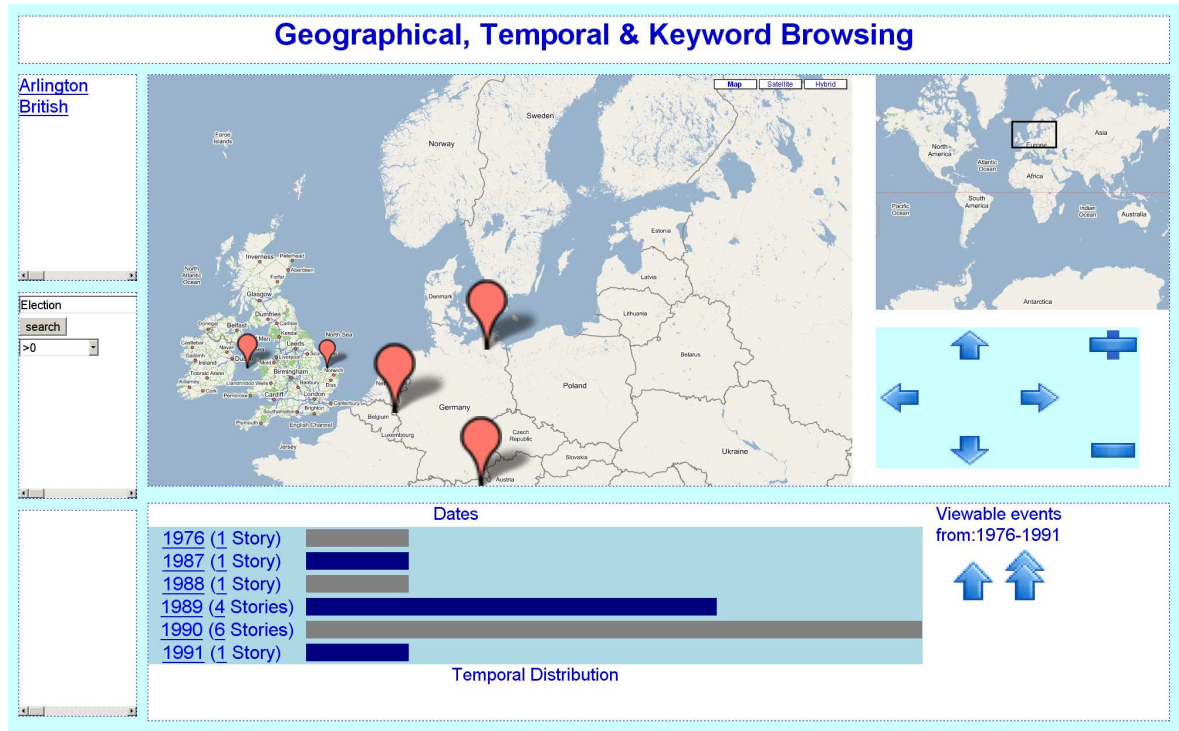


Figure 12: Geo-temporal browsing in action

semy inherently present in under-specified query-by-example scenarios, in the question how to combine possibly conflicting evidence and the responsiveness of the multimedia searches. In the last part of the chapter we have given some examples of user-centred methods that support resource discovery in multimedia digital libraries. Each of these methods can be seen as an alternative mode to the traditional digital library management tools of meta-data and classification. The new visual modes aim at generating a multi-faceted approach to present digital content: *video summaries* as succinct versions of media that otherwise would require a high bandwidth to display and considerable time by the user to assess; *information visualisation* techniques help the user to understand a large set of documents that match a query; *visual search* and *relevance feedback* afford the user novel ways to express their information need without taking recourse to verbal descriptions that are bound to be language-specific; alternative resource discovery modes such as *lateral browsing* and *geo-temporal browsing* will allow users to explore collections using lateral associations and geographic or temporal filters rather than following strict classification schemes that seem more suitable for trained librarians than the occasional user of multimedia collections. The cost for these novel approaches will be low, as they are automated rather than human-generated. It remains to be seen how best to integrate these services into traditional digital library designs and how much added value these services will bring about (Bainbridge et al 2005).

The following four books provide further reading for content-based multimedia retrieval; they complement the material of this chapter:

A del Bimbo (1999): *Visual Information Retrieval*. Morgan Kaufmann.

M Lew, ed. (2001): *Principles of Visual Information Retrieval*. Springer.

Y-J Zhang, ed. (2006): *Semantic-based visual information retrieval*. Idea Group Inc.

S Rüger (to appear 2009): *Multimedia information retrieval*. Morgan & Claypool.

Acknowledgements: The paradigms outlined in this chapter and their implementations would not have been possible without the ingenuity, imagination and hard work of all the people I am fortunate to work with or to have worked with: Paul Browne, Matthew Carey, Daniel Heesch, Peter Howarth, Partha Lal, João Magalhães, Alexander May, Simon Overell, Marcus Pickering, Adam Rae, Jonas Wolf, Lawrence Wong and Alexei Yavlinsky.

Credits: The photograph in Figure 3.3 © by Stefan Rüger, taken May 1996 in the Århus Art Museum. The screenshots in Figures 3.4 – 3.8 and 3.10 – 3.12 are reproduced courtesy of © Imperial College London. The ANSES system in Figure 3.4 was originally designed by Marcus Pickering and later modified by Lawrence Wong; the images and part of the text displayed in the screenshot of Fig 3.4 were recorded from British Broadcasting Corporation (BBC), www.bbc.co.uk. The Sammon map in Figure 3.5 and the radial visualisation in Figure 3.7 were designed by Matthew Carey. The Dendro map in Figure 3.6 was designed by Daniel Heesch. The üBase system depicted in the screenshots of Figure 3.8 (a), 3.8 (b) and 3.10 (a) was designed by Alexander May. The images used within the screenshot of Figure 3.8 and within the illustration of Figure 3.9 were reproduced from Corel Gallery 380,000, © Corel Corporation, all rights reserved. The images in the (partial) screenshots of Figures 3.10 and 3.11 were reproduced from TREC Video Retrieval Evaluation 2003 (TRECVID), <http://www-nlpir.nist.gov/projects>. The geotemporal browsing screenshot in Figure 3.12 was created by Simon Overell.

References

- C Aggarwal and P Yu (2000). The IGrid index: reversing the dimensionality curse for similarity indexing in high dimensional space. In *ACM International Conference on Knowledge Discovery and Data Mining*, pp 119–129.
- M Ankerst, D Keim and H Kriegel (1996). Circle segments: A technique for visually exploring large multidimensional data sets. In *IEEE Visualization*.
- J Aslam and M Montague (2001). Models for metasearch. In *ACM International Conference on Research and Development in Information Retrieval*, pp 276–284.
- P Au, M Carey, S Sewraz, Y Guo and S Rüger (2000). New paradigms in information visualisation. In *ACM International Conference on Research and Development in Information Retrieval*, pp 307–309.
- D Bainbridge, P Browne, P Cairns, S Rüger and L-Q Xu (2005). Managing the growth of multimedia digital content. *ERCIM News: special theme on Multimedia Informatics* 62, 16–17.
- B Bartell, G Cottrell and R Belew (1994). Automatic combination of multiple ranked retrieval systems. In *ACM International Conference on Research and Development in Information Retrieval*, pp 173–181.
- J Beis and D Lowe (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *International Conference on Computer Vision and Pattern Recognition*, pp 1000.
- W Birmingham, R Dannenberg and B Pardo (2006). Query by humming with the vocalsearch system. *Commun. ACM* 49(8), 49–52.
- D Blei and M Jordan (2003). Modeling annotated data. In *ACM International Conference on Research and Development in Information Retrieval*, pp 127–134.
- K Börner (2000). Visible threads: A smart VR interface to digital libraries. In *International*

- Symposium on Electronic Imaging 2000: Visual Data Exploration and Analysis*, pp 228–237.
- I Campbell (2000). *The ostensive model of developing information-needs*. PhD thesis, Uni of Glasgow.
- P Cano, E Batlle, T Kalker and J Haitsma (2002). A review of algorithms for audio fingerprinting. In *International Workshop on Multimedia Signal Processing*, pp 169–173.
- S Card (1996). Visualizing retrieved information: A survey. *IEEE Computer Graphics and Applications* 16(2), 63–67.
- M Carey, D Heesch and S Rüger (2003). Info navigator: a visualization interface for document searching and browsing. In *International Conference on Distributed Multimedia Systems*, pp 23–28.
- B Chawda, B Craft, P Cairns, S Rüger and D Heesch (2005). Do "attractive things work better"? An exploration of search tool visualisations. In *BCS Human-Computer Interaction Conference*, Volume 2, pp 46–51.
- M Christel and A Warmack (2001). The effect of text in storyboards for video navigation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp 1409–1412.
- M Christel, A Warmack, A Hauptmann and S Crosby (1999). Adjustable filmstrips and skims as abstractions for a digital video library. In *IEEE Forum on Research and Technology Advances in Digital Libraries*, pp 98.
- A Cockburn and J Savage (2003). Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods. In *BCS Human-Computer Interaction Conference*, pp 87–102.
- I Cox, M Miller, T Minka, T Papathomas and P Yianilos (2000). The Bayesian image retrieval system, PicHunter. *IEEE Trans on Image Processing* 9(1), 20–38.
- K Cox (1992). Information retrieval by browsing. In *Int'l Conf on New Information Technology*, pp 69–80.
- K Cox (1995). *Searching through browsing*. PhD thesis, Uni of Canberra.
- G Crane (Ed) (2005). *Perseus Digital Library Project*. Tufts Uni, 30 May 2005, <http://www.perseus.tufts.edu>.
- B Croft and T Parenty (1985). Comparison of a network structure and a database system used for document retrieval. *Information Systems* 10, 377–390.
- H Cunningham (2002). GATE, a general architecture for text engineering. *Computers and the Humanities* 36, 223–254.
- D Dearholt and R Schvaneveldt (1990). Properties of Pathfinder networks. In R Schvaneveldt (Ed), *Pathfinder associative networks: Studies in knowledge organization*, pp 1–30. Norwood.
- S van Dongen (2000). A cluster algorithm for graphs. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands.
- S Doraisamy and S Rüger (2003). Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems* 21(1), 53–70.
- P Duygulu, K Barnard, N de Freitas and D Forsyth (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European Conference on Computer Vision*, pp 97–112.

- S Feng, R Manmatha and V Lavrenko (2004). Multiple Bernoulli relevance models for image and video annotation. In *International Conference on Computer Vision and Pattern Recognition*, pp 1002–1009.
- R Fowler, B Wilson and W Fowler (1992). Information navigator: An information system using associative networks for display and retrieval. Technical Report NAG9-551, 92-1, Dept of Computer Science, University of Texas.
- D Heesch (2005). *The NN^k technique for image searching and browsing*. PhD thesis, Imperial College London.
- D Heesch, M Pickering, S Rüger and A Yavlinsky (2003). Video retrieval using search and browsing with key frames. In *TREC Video Retrieval Evaluation*.
- D Heesch and S Rüger (2003). Performance boosting with three mouse clicks — relevance feedback for CBIR. In *European Conference on Information Retrieval*, pp 363–376.
- D Heesch and S Rüger (2004). NN^k networks for content based image retrieval. In *European Conference on Information Retrieval*, pp 253–266.
- M Hemmje, C Kunkel and A Willet (1994). Lyberworld — a visualization user interface supporting fulltext retrieval. In *ACM International Conference on Research and Development in Information Retrieval*, pp 249–259.
- P Hoffman, G Grinstein and D Pinkney (1999). Dimensional anchors: A graphic primitive for multidimensional multivariate information visualizations. In *New Paradigms in Information Visualisation and Manipulation in conjunction with ACM CIKM*, pp 9–16.
- P Howarth and S Rüger (2005). Trading precision for speed: localised similarity functions. In *International Conference on Image and Video Retrieval*, pp 415–424.
- Y Ishikawa, R Subramanya and C Faloutsos (1998). MindReader: Querying databases through multiple examples. In *International Conference on Very Large Databases*, pp 218–227.
- E Izquierdo and D Djordjevic (2005). Using relevance feedback to bridge the semantic gap. In *International Workshop on Adaptive Multimedia Retrieval*, pp 19–34.
- J Jeon, V Lavrenko and R Manmatha (2003). Automatic image annotation and retrieval using cross-media relevance models. In *ACM International Conference on Research and Development in Information Retrieval*, pp 119–126.
- R Korfhage (1991). To see or not to see — is that the query? In *ACM International Conference on Research and Development in Information Retrieval*, pp 134–141.
- V Lavrenko, R Manmatha and J Jeon (2003). A model for learning the semantics of pictures. In *Neural Information Processing Systems*, pp 553–560.
- J Magalhães and S Rüger (2007). Information-theoretic semantic multimedia indexing. In *International ACM Conference on Image and Video Retrieval*, pp 619–626.
- D Metzler and R Manmatha (2004). An inference network approach to image retrieval. In *International Conference on Image and Video Retrieval*, pp 42–50.
- Y Mori, H Takahashi and R Oka (1999). Image-to-word transformation based on dividing and vector quantizing images with words. In *Int’l Workshop on Multimedia Intelligent Storage and Retrieval Management*.
- W Müller and A Henrich (2004). Faster exact histogram intersection on large data collections using inverted VA-files. In *International Conference on Image and Video Retrieval*, pp 455–463.

- S Nene and S Nayar (1997). A simple algorithm for nearest neighbor search in high dimensions. *IEEE Trans Pattern Anal Mach Intell* 19(9), 989–1003.
- OCLC (2008). Online computer library center, worldcat: About: Worldcat facts and statistics: Worldcat statistics by format as of Mar 2008. <http://www.oclc.org/worldcat/statistics> (accessed Mar 2008).
- M Pickering (2004). *Video Retrieval and Summarisation*. PhD thesis, Imperial College London.
- M Pickering, L Wong and S Rüger (2003). ANSES: Summarisation of news video. In *International Conference on Information and Knowledge Management*, pp 481–486.
- K Rodden, W Basalaj, D Sinclair and K Wood (1999). Evaluating a visualization of image similarity. In *ACM International Conference on Research and Development in Information Retrieval*, pp 36–43.
- Y Rui, T Huang and S Mehrotra (1998). Relevance feedback techniques in interactive content-based image retrieval. In *Storage and Retrieval for Image and Video Databases*, pp 25–36.
- J Rydberg-Cox, L Vetter, S Rüger and D Heesch (2004). Approaching the problem of multilingual information retrieval and visualization in Greek and Latin and Old Norse texts. In *European Conference on Digital Libraries*, pp 168–178.
- J Sammon (1969). A nonlinear mapping for data structure analysis. *IEEE Trans on Computers* C-18(5), 401–409.
- S Santini and R Jain (2000). Integrated browsing and querying for image databases. *IEEE Multimedia* 7(3), 26–39.
- J Seo, J Haitisma, T Kalker and C Yoo (2004). A robust image fingerprinting system using the radon transform. *Signal Processing: Image Communication* 19, 325–339.
- J Shaw and E Fox (1994). Combination of multiple searches. In *Text Retrieval Conference*, pp 243–252.
- B Shneiderman, D Feldman, A Rose and X Ferré Grau (2000). Visualizing digital library search results with categorical and hierarchical axes. In *ACM Digital Libraries*, pp 57–66.
- A Smeaton, C Gurrin, H Lee, K Mc Donald, N Murphy, N O’Connor, D O’Sullivan, B Smyth and D Wilson (2004). The Físchlár-news-stories system: Personalised access to an archive of TV news. In *RIAO Conference on Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*, pp 3–17.
- D Squire, W Müller, H Müller and T Pun (2000). Content-based query of image databases: inspirations from text retrieval. *Pattern Recognition Letters* 21(13–14), 1193–1198.
- T Tolonen and M Karjalainen (2000). A computationally efficient multi-pitch analysis model. *IEEE Transactions on Speech and Audio Processing* 8, 708–716.
- A Torralba and A Oliva (2003). Statistics of natural image categories. *Network: Computation in Neural Systems* 14, 391–412.
- TRECVID (2003). Trec video retrieval evaluation. <http://www-nlpir.nist.gov/projects/tv2003/> last accessed Feb 2006.
- G Tzanetakis and P Cook (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10(5), 293–302.
- J Voss (2007). Tagging, folksonomy & co - renaissance of manual indexing? *Computing Research Repository abs/cs/0701072*, 1–12.

- A de Vries, N Mamoulis, N Nes and M Kersten (2002). Efficient k-nn search on vertically decomposed data. In *ACM International Conference on Management of Data*, pp 322–333.
- D Watts and S Strogatz (1998). Collective dynamics of 'small-world' networks. *Nature* 393, 440–442.
- R Weber, H-J Stock and S Blott (1998). A quantitative analysis and performance study for similarity search methods in high-dimensional space. In *International Conference on Very Large Databases*, pp 194–205.
- M Wood, N Campbell and B Thomas (1998). Iterative refinement by relevance feedback in content-based digital image retrieval. In *ACM Multimedia*, pp 13–20.
- A Yavlinsky, M Pickering, D Heesch and S Rüger (2004). A comparative study of evidence combination strategies. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp 1040–1043.
- A Yavlinsky, E Schofield and S Rüger (2005). Automated image annotation using global features and robust nonparametric density estimation. In *International Conference on Image and Video Retrieval*, pp 507–517.

Exercises

Search types

Look at the matrix of search engine types in Figure 1. Give a search and retrieval scenario for each element of this query retrieval matrix. What would be the most appropriate search technology (piggy-back text search, feature classification, content-based, fingerprint) for each scenario?









Colour histograms

Colour is perceived as a point (r, g, b) in a *three-dimensional* space. Each colour component (red, green and blue) is usually encoded as an integer between 0 and 255 (1 byte); there are two principal methods to create colour histograms: you can compute three 1d histograms of each of the r, g and b components independently or you can compute one 3d colour histogram. a) If you divide each of the red, green and blue axes into n_r, n_g, n_b equal intervals, respectively, then the 3d colour cube $[0, 255]^3$ is subdivided into $n_r n_g n_b$ cuboids or bins. Show that by mapping (r, g, b) to

$$\left\lfloor \frac{n_r r}{256} \right\rfloor n_g n_b + \left\lfloor \frac{n_g g}{256} \right\rfloor n_b + \left\lfloor \frac{n_b b}{256} \right\rfloor$$

you get an enumeration scheme $0, \dots, n_r n_g n_b - 1$ for the cuboids. A 3d colour histogram of an image is a *list of the numbers of the pixels in an image that fall into each of the different cuboids* in colour space. In other words, you look at each pixel in an image, compute above index from its colour (r, g, b) and increment the corresponding variable, which records the number of pixels that fall into this colour cuboid.

b) Compute both types of colour histograms for the image below (the colours of the stripes are given by the table next to the image). Use $64 = 4^3$ bin cubes for the 3d-bin-histogram and 22 bins for each of the three colour-component histograms (yielding 66 bins altogether), so that both histogram types have a similar number of bins.

	R	G	B	
	0	0	0	black
	255	0	0	red
	0	255	0	green
	0	0	255	blue
	0	255	255	cyan
	255	0	255	magenta
	255	255	0	yellow
	255	255	255	white

c) Which of the two colour histogram methods has retained more information about the colour distribution in the original picture? How came it about that one of the two methods lost vital information despite having roughly the same number of bins (64 vs 66)?

d) Why are usually *normalised* histograms computed and stored for content-based retrieval? Normalised histograms store the proportions of pixels in the respective bins not the absolute number of pixels.

Image search

Sketch the block diagram of a colour-and-texture-based image search engine for curtain fabrics. Explain the general workings of a content-based search engine and contrast it with the workings of a text search engine in terms of retrieval and indexing technology.

Indicative solutions

Search types

It is relatively easy to come up with a usage scenario for each of the matrix elements in Figure 1: for example, the image input speech output matrix element might be “given an X-ray image of a patient’s chest, retrieve dictaphone documents with a relevant spoken description of a matching diagnosis”. However, creating satisfying retrieval solutions is highly non-trivial and the main subject of the multimedia information retrieval discipline.

In essence, Figure 1 lists cross-modal retrieval scenarios. The left column describes situations where the search is by text, for example, using a web-search-engine type-interface such as the one in Figure 4 for TV news. Users have grown accustomed to this search method, and it can easily be implemented as long as the document repositories (be it text, video, images, speech, music, sketches etc) have textual meta-data associated with their individual entries. Museum catalogues would be good examples of sources of these meta-data. Depending on the document types in the repository at hand it may be more or less easy to extract a surrogate text of the original document. This is almost trivial when the repository consists of text documents: One would strip these from formatting instructions and, for example, stem words in order to remove variations brought about by grammar rules, to arrive at a “bag of words” that can be indexed. The audio track of video or speech documents could undergo automated speech recognition; one would hope that redundancy of word repetitions in the original document somehow alleviates the effect of transcription errors.

In the case of “query by example” modes, where the query is of the same type as the document in the repository, one immediate approach would be relying on a type of content-based or fingerprinting retrieval. The trick here is to be able to extract features that are specific to the document and invariant under perceptually or otherwise irrelevant transformations of the query or the document: it should not matter whether spoken queries are issued by a man or a woman, whether an image as query input is scaled, has undergone a high or low image compression, whether a hummed query is slightly higher or lower in pitch, etc. This requirement is non-trivial: eg, for medical image retrieval the features should pick up only medically relevant aspects of the imaging, and this normally requires much domain expertise and arguably a fair amount of signal processing skills to achieve that.

In the case of retrieving speech documents against spoken queries the common best practice appears to be *not* to use low-level feature matching of the processed speech signal, but instead transcribe both queries and speech documents to text. In contrast, approaches to query by humming — a thriving challenge of the growing music information retrieval community — most commonly deploy low-level features for the matching process. Currently, it is not clear what the best approach for image search by image similarity would be: low-level feature matching or automated annotation. In the first case common research questions are which features and which similarities to use, while the second case requires an appropriate choice of symbolic vocabulary and suitable models, typically from machine learning, to facilitate the automated annotation.

The level of desired similarity or sameness is also an important factor in deciding which techniques are more likely to succeed. In order to locate the exact performance of, say, a music piece that you are listening to on the radio, one would likely employ a signal-based fingerprinting technique that is invariant to distortions brought about by the broadcasting and recording (eg, invariant to the background noise in a car when recording the music sample with a mobile phone) but still sensitive to the qualities of the particular performance. If on the other hand, one was interested more generally in versions of the same melody or lyrics by possibly different artists performed within possibly even different genres (a pop version of a jazz song, say) then a more symbolic or coarse feature extraction process may be more

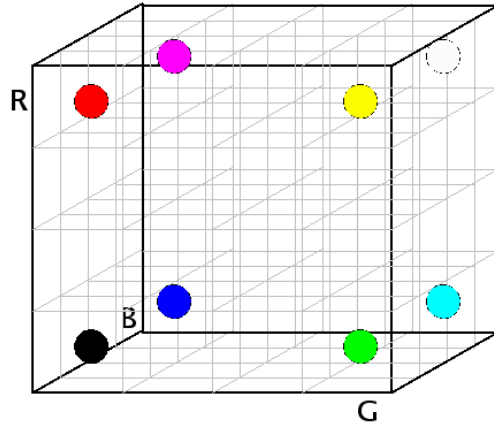
appropriate.

Cross-modal retrieval where queries and documents are of *different* type clearly require some sort of translation, which in the simplest case could be a textual annotation of either using a discrete set of vocabulary. The cross-modal translation is the most involved, and least explored, of all the cases in Figure 1.

Colour histograms

a) The first thing to note is that $r/256$ is smaller than 1, as r runs from 0 to 255; hence, $i_r := \lfloor n_r r / 256 \rfloor$ is an integer that takes on values from 0 (for $r = 0$) to $n_r - 1$. Note that the floor function $\lfloor x \rfloor$ returns the largest integer smaller than or equal to x . Similarly, $i_g := \lfloor n_g g / 256 \rfloor \in \{0, \dots, n_g - 1\}$ and $i_b := \lfloor n_b b / 256 \rfloor \in \{0, \dots, n_b - 1\}$. The 3d bins in colour space correspond to a 3d array that has the dimensionality $\text{bin}[n_r][n_g][n_b]$. i_r , i_g and i_b are valid indices for this 3d array. A compiler might assign the location $i_r n_g n_b + i_g n_b + i_b$ in an equivalent one-dimensional array; this assumes that array indices start from 0 and that the last index of a multidimensional array addresses neighbouring locations in memory as is the case, eg, in the programming language C++.

b) Of the 64 different 3d bins (numbered $0, \dots, 63$) in above scheme only eight bins are occupied, as there are only 8 different colours used in the example picture: bin 0 (black), 3 (blue), 12 (green), 15 (cyan), 48 (red), 51 (magenta), 60 (yellow) and 63 (white). The following figure visualises this 3d colour histogram:



The area of the circles in the occupied bins indicate how much this bin is occupied; the colour corresponds to average colour that this bin summarises.

In contrast, the three 1d histograms are identical: The values $0, \dots, 255$ are subdivided into 22 different bins, and bin 0 and bin 21 are the only ones that are occupied in each of the r , g and b histograms.

c) The 3d colour histogram has retained more information about the colour use in the original image than the three 1d histograms despite the fact that there are more bins in the three 1d histograms than in the 3d histogram (66 vs 64). The 3d histogram captures the *joint distribution* of the r , g and b values and clearly recognises that $1/8$ th of the pixels are black, blue, green, cyan, red, magenta, yellow and white, respectively. The three 1d histograms have only retained the *marginal distribution* of the r , g and b values. As such images with a completely different colour distribution can have the same marginal distributions. In fact, a black and white image with half the pixels black and the other half of the pixels white would have the same marginal distributions. Marginalisation of multi-dimensional distributions always loses information.

d) Normalising histograms makes them scaling invariant, ie, larger and smaller versions of the same image result in the same histogram.

Image search

Figure 2 can be modified to cater for the colour-and-texture-based image search engine at hand. The query would be an example image containing curtain fabrics of similar colour and texture. The feature extraction process would deploy colour histograms, for example, the 3d colour histogram from the previous exercise and a texture feature vector. Both vectors could be concatenated into a single vector representation. In the end the query image will have a certain representation as a point in feature space and so will every single image in the database.

The images whose representations are closest to the representation of the query are ranked top by this process. In fact, all images in the database will be sorted by their distance in feature space to the query image. Thus the feature representation and distance notion in feature space are key aspects of the workings of this type of content-based search engine. A very simple distance of the query feature vector f^q to an arbitrary image feature vector f^i is the Manhattan distance

$$d(f^q, f^i) = \sum_{j=1}^n |f_j^q - f_j^i|,$$

which is simply the sum of the absolute differences of the respective n components of the feature vector.

Index

- Annotation
 - automated, 2–4, 9
- ANSES, 8, 9
- Browsing, 14
 - geotemporal, 17
 - lateral, 14–17
- Catalogue, 1
- Classification, 1, 2, 4
 - music genres, 4
- Content-based retrieval, 2, 4–6
- Distance measurement, 5
- Feedback
 - relevance, 5, 7, 13, 14
- Fingerprinting
 - audio, 6
- Fusion problem, 7
- Indexing
 - multimedia, 1–3, 5–7, 24
- Information visualisation, 9–11, 13
- Keyframes, 8
- Lateral neighbours, 14
- Meta-data, 1, 2, 6, 13
- MIDI, 3
- Multimedia indexing, 1–3, 5–7, 24
- Pathfinder networks, 14
- Piggy-back retrieval, 2, 3, 24
- Query point moving, 13
- Query-by-Example, 2, 4, 5
- Query-by-Humming, 3
- Relevance feedback, 5, 7, 13, 14
- Resource discovery, 1, 2
- Responsiveness problem, 7
- Semantic gap, 1, 6, 7
- Storyboards, 8
- Subtitles, 3, 9
- Video summaries, 7, 9
- Visualisation
 - Dendro map, 10
 - radial, 11
 - Sammon map, 9, 10